**Cleaning Data with OpenRefine**

**What is OpenRefine?**

* Free data cleaning tool that used to be run by google (Google Refine)
  + detecting and fixing inconsistencies
  + transforming data from one structure or format to another
* General procedures
  + 1) Create a project
  + 2) Import the data
  + 3) Manipulate the data with facets, clusters and edits
  + 4) Export the data in your favorite format
* An OpenRefine project consists of one table.  All actions that were done on a dataset are stored in a project and can be replayed on another dataset.
* Unlike spreadsheets, no formulas are stored in the cells, but formulas are used to transform the data, and transformation is done only once.
* OpenRefinemaxes out at around 60,000 records when it starts have trouble

**Launching OpenRefine**

* When you open the program, it should launch a browser on your computer
* If it doesn‘t, open a new browser yourself and “connect“ to your computer which is hosting the program: [http://127.0.0.1:3333](http://127.0.0.1:3333/)

**Import data into OpenRefine**

* Before you load something in Refine, look in Excel first to make sure it’s what you want
* In Excel, you can create an original sort order which is a unique ID in case you screw something up later.
* Can import CSV files, Excel files and google spreadsheets.
* Create a new project and import **lotterwinners.xlsx**. Make sure the right options are selected before you create the project.
* Press “create project.”

**Basic functions of OpenRefine**

* This is the heart of Refine, where you do the work. (show elements of this screen)
  + Keep an eye on the number of **rows of data** you have because this will change
  + Note that you can only see a **max of 50 rows** at a time and have to scroll through them.
  + **“Rows” versus “records”:** When you first open the file, pay attention to whether the data shows up as "Records" or "Rows". The difference is that any item that has nothing in the first column is assumed to be part of the record above it. The first column determines whether it's combined.
  + Go to the arrow next to the column.
    - Under **“edit cells”,** you can fill down or transform your data in common ways (remove white spaces, convert the format of the cells)
    - Under **“edit columns”,** you can rename the column or reorder the columns.
    - You can **“transpose”** your data so columns are rows and rows are columns.
    - Under **“view”,** you can change which columns you’re viewing
  + “Facet/filter” tab will show you what filters you’ve applied to your data
  + “Undo/redo” tab has an infinite undo list!! (unlike other programs)
  + Export button will let you export your final dataset in different file types
* Let’s copy our column. Creating a new column let us compare what our cells look like before and after we clean them.
  + Hit the drop-down on “w\_town”

“Edit Column” then “Add column based on this column”

* + Name it “town” and click OK
  + In orange at the top, it says what just happened and tells us that “undo” is possible.
* Show Sort function
  + Drop-down, “Sort…” | “Sort” (and mention various options)
* Show Text Filter function
  + Drop-down | “text filter” | use box at left
* But you didn’t come here to learn an Excel alternative. Here comes the real deal..

**Cleaning Data in OpenRefine**

* “Facet” function
  + Click the drop-down on Town and select “Facet” and “Text Facet”. See on the left we get something that’s like a pivot table grouped by Town. But in this case, we can change what’s in our pivot table which changes the original data.
  + Scroll down and see we have “Blomfield” and “Bloomfield”.
    - Edit “Blomfield” to be “Bloomfield”. See that they collapse into one piece of information. We can do this for other names too, but it’s going to take a while.
* “Cluster” function
  + We can fix abbreviations and bad spelling with “clusters,” which are algorithms that group information in cells together based on common letters or sounds. You can decide whether the algorithm found a good match, and if so, change the underlying data.
  + Click “Cluster”
  + Notice that “E. Orange” and “E Orange” are clearly the same.
  + Click “merge” and click on the preferred name out of the two, or type something new, like “East Orange” (“Wood Ridge” is OK. Those are different towns.)
  + Click “Merge selected and recluster” and repeat as many times as necessary. You will potentially find new clusters after merging some together.
  + **NEVER** **press select all** and just recluster. Browse through to make sure it did it right.
  + Using **“Key Collision”** method, switch the “keying function”:
    - **Fingerprint:** looks for identical characters, e.g. “John Smith” and “Smith John”. It will tildes, periods, duplicate words and make everything lower case).
    - **Ngram-fingerprint**. The higher the N, the more variations allowed, meaning the greater the differences between the words and the match. Might go as high and 5 or 6 but you’ll run into more problems. Start small and keep bumping it up until they don’t make sense anymore.
    - **Metaphone3:** looks for similar sounds like “smith” and “smyth”. Based on English words.
      * On the right, see the **historgrams** – choose how many choices you want to see in clusters or the number of rows in a cluster you want to look at
    - **Cologne-phonetic:** looks for similar sounds like “smith” and “smyth”. Based on Germanic words so doesn’t work as well.
      * On the right, see the **historgrams** – choose how many choices you want to see in clusters or the number of rows in a cluster you want to look at
  + Switch method to **“Nearest Neighbor”:** looking for clusters of characters that are identical (“Fred” and “Frederick”). This takes longer to run, especially on bigger files.
    - **Levenshtein:** the number of edits it would take to get from one to another (“Roberto Rocha” to “robertorocha” is 3). The radius is how far you’re looking from one letter to another. Don’t mess around too much here.
    - **PPM:** looks at partial matches (and can also increase the radius)

**Exporting Data in OpenRefine**

* We’re done cleaning this file. Let’s export it.
  + Click “Export” and select a file type.
  + Now you’ve got a cleaned version for what you want to do next

**Let’s try another example**

* Import **nj-hospital-2012.xlsx** into OpenRefine
* Yuck -- look at the rows! Let’s split them.
  + Drop-down “Edit columns” | “split into several columns” (first choice). Change the separator to a semicolon.
  + Uncheck “remove this column”!!!
  + Oops! Look at the zip codes. They are only 4 characters. Why? they’re in NJ, which has zip codes that start with zeros.
  + Undo: Click “undo/redo” and step 0. We’re now back at that step.
  + Try it again
    - “Edit columns” | “split into several columns” (first choice)
    - uncheck “remove this column”
  + uncheck “guess cell type”
  + It’s good!
* Hospital names are inconsistent. Let’s standardize them
  + Drop-down: “Edit Cells” | “Common transformation” | “to titlecase”
* Let’s change the column names too.
  + Drop-down: “Edit Column” | “Rename this column” | rename to “Hospital Name”, etc.
* Export and you’ve got the dataset.

**Let’s try one more example**

* Create project about Political Contributions (looking for Bruno, majority leader of the senate)
  + Open” | “create project” | “choose files”
  + Select “Nassau police union contributions” and “next”
  + “Create project”
* Let’s review clustering
  + City drop-down | “facet” | “text facet”
    - 240 choices
  + City | “Edit cells” | “Common tranformations” | “trim leading and trailing whitespaces”
    - 238 choices
  + In the City text facet (in the panel on left), click “cluster”
  + “Merge them all and recluster” then “close”
* Let’s filter
  + City drop-down
  + Type “Albany”
* Name text facet
  + Click “include” next to anything that mentions Bruno
  + Click “cluster”
  + Change method to “nearest neighbor” and distance function to “PPM”
  + Change all to the same thing by typing it and pasting it for each

Advanced OpenRefine with Regular Expressions: <https://github.com/sarahcnyt/data-journalism/tree/master/openrefine>

*Adapted from a NICAR 2014 presentation by Frederick Kaimann*